```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
cd /content/drive/MyDrive
```

```
/content/drive/MyDrive
```

```
#unzip
```

```
!unzip Flowers-Dataset.zip
```

```
unzip:  cannot find or open Flowers-Dataset.zip, Flowers-
Dataset.zip.zip or Flowers-Dataset.zip.ZIP.
```

```
#import
```

```
import warnings
warnings.filterwarnings('ignore')
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Importing req. lib
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
#Image augmentation
```

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
```

```
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/
flowers',target_size=(64,64),class_mode="categorical",batch_size=24)
```

```
Found 4317 images belonging to 5 classes.
```

```
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/
flowers',target_size=(64,64),class_mode="categorical",batch_size=24)
```

```
Found 4317 images belonging to 5 classes.
```

```
x_train.class_indices
```

```
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

```
# Import req. lib

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D,
Flatten, Dense

#Create model

model = Sequential()

#Convolution layer

model.add(Convolution2D(32,
(3,3),activation='relu',input_shape=(64,64,3)))

#Maxpooling

model.add(MaxPooling2D(pool_size=(2,2)))
```

```
#Flatten

model.add(Flatten())

#Output

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896       
 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

```
32*(3*3*3+1)
```

```
896
```

```
#Denselayer
```

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
```

```
#Output
```

```
model.add(Dense(5, activation='softmax'))
```

```
# Compiling the model
```

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics
=['accuracy'])
```

```
len(x_train)
```

```
180
```

```
#Save the model
```

```
model.save('flowers.h5')
```

```
#Test model
```

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
model=load_model('flowers.h5')
```

```
img=image.load_img(r'/content/drive/MyDrive/flowers/sunflower/
10386503264_e05387e1f7_m.jpg')
```

```
img
```

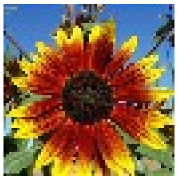
```
x=image.img_to_array(img)
```

```
x
```

```
array([[[144., 125.,   5.],
        [108., 128.,  41.],
        [123., 133.,  21.],
        ...,
        [121., 133.,  21.],
        [128., 139.,  35.],
        [ 86.,  90.,   3.]],
       [[113., 148.,   2.],
        [ 37.,  93., 242.],
        [ 41.,  90., 227.],
        ...,
        [ 58., 115., 231.],
        [ 68., 120., 242.],
        [ 51.,  62.,  54.]],

       [[149., 143.,  21.],
        [ 44.,  95., 220.],
        [ 55., 105., 190.],
        ...,
        [ 77., 118., 197.],
        [ 71., 130., 248.],
        [ 55.,  66.,  36.]],

       ...,

       [[141., 145.,   6.],
        [  2.,  13.,  15.],
        [ 27.,  41.,  24.],
        ...,
        [110., 151.,  75.],
        [128., 171., 100.],
        [ 53.,  66.,  12.]],

       [[149., 151.,  24.],
        [  0.,  10.,   6.],
        [ 11.,  24.,   0.],
        ...,
        [129., 172., 100.],
        [138., 184., 119.],
        [ 67.,  85.,  35.]],

       [[102.,  97.,   3.],
        [ 33.,  41.,  20.],
        [ 53.,  61.,  12.],
        ...,
        [ 34.,  48.,   0.],
        [ 43.,  61.,  13.],
        [ 35.,  24.,   0.]]], dtype=float32)
```

```
img=image.load_img(r'/content/drive/MyDrive/flowers/sunflower/
10386503264_e05387e1f7_m.jpg',target_size=(64,64))
```

```
img
```



```
x_train.class_indices
```

```
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

```
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
```

```
pred_prob=model.predict(x)
```

```
class_name=["Daisy","Dandelion","Rose","Sunflower","Tulip"]
```

```
pred_id=pred_prob.argmax(axis=1)[0]
pred_id
```

```
0
```

```
print("Predicted flower is",str(class_name[pred_id]))
```

```
Predicted flower is Daisy
```